UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/537,430	06/02/2005	Tim Neil	93422-46	3156

22463 7590 11/26/2008

SMART AND BIGGAR
438 UNIVERSITY AVENUE
SUITE 1500 BOX 111
TORONTO, ON M5G2K8
CANADA

EXAMINER	
TURNER, ASHLEY D	

ART UNIT	PAPER NUMBER
2454	

MAIL DATE	DELIVERY MODE
11/26/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/537,430	**Applicant(s)** NEIL ET AL.	
	Examiner ASHLEY D. TURNER	**Art Unit** 2454	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 September 2008*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *2-14 and 17-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *2-14 and 17-21* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date *6/2/2005,11/3/2005*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 2-14,17-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mikhailov (US 6,990534 B2) in view of Fascenda (US 6,466,937 B1).

Regarding claim 2

Referring to claim 2 Mikhailov discloses all the limitations of claim 2 which is described above. Mikhailov also discloses queuing said given message on said a queue for said application; and subsequent to said queuing, pushing said given message, and each message queued on said queue, toward a destination for said application of said application server, and wherein said pushing comprises, for each message on said queue, dequeuing said each message from said queue and pushing said each message. (Col. 61 Fig. 61 FIG. 34A is a block diagram illustrating the invented server-initiated content delivery process (see also FIG. 35A). The delivery request originates at the application server 25005, 25007, in response to application algorithms, that can be defined by those skilled in art at application design time. The request then follows to Messaging Systems 25003, 25008 to the messaging queue or topic that the application and PES

27 negotiated for content delivery requests. In this architecture the Application Server 25005, 25007 is the publisher of the messages to the messaging topics or queues 25012 and the PES 27 is the subscriber for the messages through message subscription(s) 25004. When the message arrives to the PES 27 it is received by Push Interface 1014, which provides bridging and decoding functions between Messaging System 25003, 25008 and PES 27 components. Once the message is processed the information follows to the Content Manager 1021, which fetches content delivery request data from the Application Servers 25005, 25007 or any other content provider using HTTP requests through Web Server 24 or other applicable means. Upon it receiving the data PES 27, preprocesses and validates it, and saves the content to the Queue 46, which stores the content in Queue storage 1020. In the next step the Content Manager 1021 communicates to the router 1017, which reads information from Routing tables 47, to decide if the device is available and can process server-initiated content delivery. Once the device is available, the content is sent to the PAT 22 through one or more gateways 25001, 25002 and Mobile Networks 25009, 25010. At this point the device receives the content and follows the algorithms described above to present the content to the user. During content delivery queuing and other related delivery processing Push Manager 1012 generates messages with delivery status notifications on each status change and publishes them through the Push Interface 1014 to messaging Queues or Topics 25012 in Messaging System 25003, 25008. Application Server 25005, 25007 can subscribe to the queues and topics through Subscriptions 25004 in order to obtain delivery status notifications if they are required by the application algorithms. Currently the PES 27 generates the messages for the following delivery events: The content is placed in the queue; The content is replaced in the queue (older content was suppressed to ensure delivery of

the most fresh content only); The content delivery failed (with attempt number and error code), and delivery attempts will continue until queue for the frame expires; The content is delivered to the target device; The content expired (queue for the frame may be reset upon expiration of application-configured timeout) . Mikhailov did not disclose a method of enabling use of an application server application by wireless communication device comprising, at a transaction server: on receipt of a given message from said wireless communication device for said application on said application server. The general concept of a method of enabling use of an application server application by wireless communication device comprising, at a transaction server: on receipt of a given message from said wireless communication device for said application on said application server, is well known in the art as taught by Fascenda. Fascenda discloses a method of enabling use of an application server application by wireless communication device comprising, at a transaction server: on receipt of a given message from said wireless communication device for said application on said application server, (Col. 10 lines 62-67 and Col. 11 lines 1-21 Sever template database 330 includes the latest or most current versions of all of the available templates in the system of the present invention. Server template database 330 also includes tables mapping individual client device unique identifiers to the most current template versions authorized for client devices 108 associated with the unique identifiers. As new services, features and options are added to the system of the present invention, new templates are stored in server template database 330 and/or existing stored templates are updated, to reflect the additions. Therefore, at any given time, it is possible a client device 108 includes an old version of a template, that is, an out-of-date template that requires updating. When server 114 receives information request message 316 from client device 108, server 114 determines whether the template (at client device 108) associated with the request message is the most current template (for example, an updated template). If the template is an old version, server 114 retrieves the most current template from server template database 330, and then transmits an appropriate template update, along with the requested information, to client device 108 using response message 318. In this manner, server 114 distributes the most current template versions to client devices 108 on a per access and an as needed basis. Thus, server 114 efficiently distributes template updates to client devices 108 to render new service features and options available to the users, and maintain configuration control over the clients.) It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Mikhialov to include a method of enabling use of an application server application by wireless communication device comprising, at a transaction server: on receipt of a given message from said wireless communication device for said application on said application server in order to provide the user with a mechanism for retrieving the retained response information and to display such information as necessary.

Regarding claim 3

Referring to claim 3 Mikhailov and Fascenda discloses all the limitations of claim 3 which is described above. Mikhailov also discloses prior to said dequeuing and pushing, acquiring a lock for said destination on said application server, said lock preventing other use of said destination. (Col.23 lines 61-67 and Col. 25 lines 1-3 If field worker accepts the work order, the "YES" branch is followed to routine 2046, in which the application updates the work order information, and locks it so that multiple field workers are not assigned to perform the same work. Then the application enters idle state until the application receives work order completion notification from the field worker; it follows to routine 2047, in which the application updates work order information according to the response from the field worker, and follows to routine 2048. In routine 2048 the system may send status reports to the field worker. Routine 2048 is followed by the "END" step, which concludes routine 2020.)

Regarding claim 4

Referring to claim 4 Mikhailov and Fascenda discloses all the limitations of claim 4 which is described above. Mikhailov also discloses after said dequeuing said each message from said queue and pushing said each message, releasing said lock for said destination on said application server. (Col.23 lines 61-67 and Col. 25 lines 1-3 If field worker accepts the work order, the "YES" branch is followed to routine 2046, in which the application updates the work order information, and locks it so that multiple field workers are not assigned to perform the same work. Then the application enters idle state until the application receives work order completion notification from the field worker; it follows to routine 2047, in which the application updates work order information according to the response from the field worker, and follows to routine

2048. In routine 2048 the system may send status reports to the field worker. Routine 2048 is followed by the "END" step, which concludes routine 2020.)

Regarding claim 5

Referring to claim 5 Mikhailov and Fascenda discloses all the limitations of claim 5 which is described above. Mikhailov also discloses wherein messages on said queue are queued on a first in first out (FIFO) basis and wherein a trailing message in said queue is not pushed until a message in said queue immediately preceding said trailing message is considered to have successfully reached said destination (Col. 61 Fig. 61 FIG. 34A is a block diagram illustrating the invented server-initiated content delivery process (see also FIG. 35A). The delivery request originates at the application server 25005, 25007, in response to application algorithms, that can be defined by those skilled in art at application design time. The request then follows to Messaging Systems 25003, 25008 to the messaging queue or topic that the application and PES 27 negotiated for content delivery requests. In this architecture the Application Server 25005, 25007 is the publisher of the messages to the messaging topics or queues 25012 and the PES 27 is the subscriber for the messages through message subscription(s) 25004. When the message arrives to the PES 27 it is received by Push Interface 1014, which provides bridging and decoding functions between Messaging System 25003, 25008 and PES 27 components. Once the message is processed the information follows to the Content Manager 1021, which fetches content delivery request data from the Application Servers 25005, 25007 or any other content provider using HTTP requests through Web Server 24 or other applicable means. Upon it receiving the data PES 27, preprocesses and validates it, and saves the content to the Queue 46, which stores the content in Queue storage 1020. In the next step the Content Manager 1021 communicates to the router 1017, which reads information from Routing tables 47, to decide if the device is available and can process server-initiated content delivery. Once the device is available, the content is sent to the PAT 22 through one or more gateways 25001, 25002 and Mobile Networks 25009, 25010. At this point the device receives the content and follows the algorithms described above to present the content to the user. During content delivery queuing and other related delivery processing Push Manager 1012 generates messages with delivery status notifications on each status change and publishes them through the Push Interface 1014 to messaging Queues or Topics 25012 in Messaging System 25003, 25008. Application Server 25005, 25007 can subscribe to the queues and topics through Subscriptions 25004 in order to obtain delivery status notifications if they are required by the application algorithms. Currently the PES 27 generates the messages for the following delivery events: The content is placed in the queue; The content is replaced in the queue (older content was suppressed to ensure delivery of the most fresh content only); The content delivery failed (with attempt number and error code), and delivery attempts will continue until queue for the frame expires; The content is delivered to the target device; The content expired (queue for the frame may be reset upon expiration of application-configured timeout)

Regarding claim 6

Referring to claim 6 Mikhailov and Fascenda discloses all the limitations of claim 6 which is described above. Mikhailov also discloses if a particular message pushed toward said destination does not successfully reach said destination, ceasing said dequeuing and pushing and re-queuing said particular message on said queue. (Col.62 Fig.35B lines 39-55 FIG. 35B is a logic flow diagram illustrating content loading routine 26004 of the server-initiated content delivery process 26000. This routine is typically implemented by the content manager 1021 in the PES 27. Routine 26004 starts with routine 26030, in which the content manager 1021 receives request to download certain content for the server-initiated content delivery request. Routine 26030 follows to routine 26031 in which the content manager 1021 loads the content from the URL read from the content delivery request and follows to the step 25033, in which it checks whether loading was successful. If loading was not successful, the "NO" branch is followed to routine 26043, in which the PES 27 sends the delivery failed message to the application server, which initiated content delivery request, through Messaging System 25008, 25003. If loading was successful, the "YES" branch is followed to the step 25035, in which the server checks if the content is of supported type. Here supported type is the type, which can be transformed by the gateway 23 to the type understood by the PAT 22.)

Regarding claim 7

Referring to claim 7 Mikhailov and Fascenda discloses all the limitations of claim 7 which is described above. Mikhailov also discloses on dequeuing said each message and prior to pushing said each message (Col. 61 Fig. 61 FIG. 34A is a block diagram illustrating the invented server-initiated content delivery process (see also FIG. 35A). The delivery request originates at the application server 25005, 25007, in response to application algorithms that can be defined by those skilled in art at application design time. The request then follows to Messaging Systems 25003, 25008 to the messaging queue or topic that the application and PES 27 negotiated for content delivery requests. In this architecture the Application Server 25005, 25007 is the publisher of the messages to the messaging topics or queues 25012 and the PES 27 is the subscriber for the messages through message subscription(s) 25004. When the message arrives to the PES 27 it is received by Push Interface 1014, which provides bridging and decoding functions between Messaging System 25003, 25008 and PES 27 components. Once the message is processed the information follows to the Content Manager 1021, which fetches content delivery request data from the Application Servers 25005, 25007 or any other content provider using HTTP requests through Web Server 24 or other applicable means. Upon it receiving the data PES 27, preprocesses and validates it, and saves the content to the Queue 46, which stores the content in Queue storage 1020. In the next step the Content Manager 1021 communicates to the router 1017, which reads information from Routing tables 47, to decide if the device is available and can process server-initiated content delivery. Once the device is available, the content is sent to the PAT 22 through one or more gateways 25001, 25002 and Mobile Networks 25009, 25010. At this point the device receives the content and follows the algorithms described above to present the content to the user. During content delivery queuing and other related delivery processing Push Manager 1012 generates messages with delivery status notifications on each status change and publishes them through the Push Interface 1014 to messaging Queues or Topics 25012 in Messaging System 25003, 25008. Application Server 25005, 25007 can subscribe to the queues and topics through Subscriptions 25004 in order to obtain delivery status notifications if they are required by the application algorithms. Currently the PES 27 generates the messages for the following delivery events: The content is placed in the queue; The content is replaced in the queue (older content was suppressed to ensure delivery of the most fresh content only); The content delivery failed (with attempt number and error code), and delivery attempts will continue until queue for the frame expires; The content is delivered to the target device; The content expired (queue for the frame may be reset upon expiration of application-configured timeout), logging said event and wherein said re-queuing said particular message comprises utilizing said log to identify messages to re-queue. (Col.22 lines 45-50 Logging and Inventory Management Engine (LIME) 1045, is a logging component, which collects and logs PAT-generated errors and warnings for inspection by administrators and application developers; Distributed Log 1046, is a non-volatile storage used to store information on errors, warnings, messages, and inventory data. It may be implemented as databases, files, etc.)

Regarding claim 8

Referring to claim 8 Mikhailov and Fascenda discloses all the limitations of claim 8 which is described above. Mikhailov also discloses timing a retry interval and, expiry of said retry interval, for each message on said queue: dequeuing said each message from said queue and pushing said each message from said queue and pushing said each message toward said destination for said application of said server. (Col. 61 Fig. 61 FIG. 34A is a block diagram illustrating the invented server-initiated content delivery process (see also FIG. 35A). The delivery request originates at the application server 25005, 25007, in response to application algorithms, that can be defined by those skilled in art at application design time. The request then follows to Messaging Systems 25003, 25008 to the messaging queue or topic that the application and PES 27 negotiated for content delivery requests. In this architecture the Application Server 25005, 25007 is the publisher of the messages to the messaging topics or queues 25012 and the PES 27 is the subscriber for the messages through message subscription(s) 25004. When the message arrives to the PES 27 it is received by Push Interface 1014, which provides bridging and decoding functions between Messaging System 25003, 25008 and PES 27 components. Once the message is processed the information follows to the Content Manager 1021, which fetches content delivery request data from the Application Servers 25005, 25007 or any other content provider using HTTP requests through Web Server 24 or other applicable means. Upon it receiving the data PES 27, preprocesses and validates it, and saves the content to the Queue 46, which stores the content in Queue storage 1020. In the next step the Content Manager 1021 communicates to the router 1017, which reads information from Routing tables 47, to decide if the device is available and can process server-initiated content delivery. Once the device is available, the content is sent to the PAT 22 through one or more gateways 25001, 25002 and Mobile Networks 25009, 25010. At this point the device receives the content and follows the algorithms described above to present the content to the user. During content delivery queuing and other related delivery processing Push Manager 1012 generates messages with delivery status notifications on each status change and publishes them through the Push Interface 1014 to messaging Queues or Topics 25012 in Messaging System 25003, 25008. Application Server 25005, 25007 can subscribe to the queues and topics through Subscriptions 25004 in order to obtain delivery status notifications if they are required by the application algorithms. Currently the PES 27 generates the messages for the following delivery events: The content is placed in the queue; The content is replaced in the queue (older content was suppressed to ensure delivery of the most fresh content only); The content delivery failed (with attempt number and error code), and delivery attempts will continue until queue for the frame expires; The content is delivered to the target device; The content expired (queue for the frame may be reset upon expiration of application-configured timeout),

Regarding claim 9

Referring to claim 9 Mikhailov and Fascenda discloses all the limitations of claim 9 which is described above. Mikhailov also discloses wherein said destination is a Component Object Model interface, a Distributed Component Object Model interface, Simple Object Access Protocol interface, a .NET interface, or a .NETRemoting interface. (Col.21 lines 32-53 The server system 1099 may include the following components: Proactivity Enablement Server (PES) 27, is a server software system, which facilitates server-initiated content delivery to mobile devices and is specially tailored for development and deployment of proactive wireless applications; Queue Storage 1020, is a supplement to PES, which stores content delivery request data for PES. It may be implemented using databases, files, etc.; Web Server 24, is a standard HTTP server or any other suitable application system, which is serving content requests and/or delivers information in the fixed network (Internet, Intranet, etc); Messaging System 1007, is a supplementary component, which function is to enable routing and delivery of digital messages in heterogeneous computing environment using well-defined messaging protocols. Examples of such messaging products include but not limited to Java Messaging Server (JMS) implementations, Microsoft MSMQ, Web Services SOAP protocol, and so forth; Application Server 25, is a system, which handles application logic and related code for proactive data applications 49 and submits content delivery requests to the Messaging System 1007 (L1); Database 26, is a component, which function is to store and manage application and other related data; other unlisted components, may be required to enable or extend functionality and/or performance of the system; PES 27 consists of the following modules).

Regarding claim 10

Referring to claim 10 Mikhailov and Fascenda discloses all the limitations of claim 10 which is described above. Mikhailov also discloses wherein said acquiring a lock comprises sending a lock request to a remote lock server. (Col.23 lines 61-67 and Col. 25 lines 1-3 If field worker accepts the work order, the "YES" branch is followed to routine 2046, in which the application updates the work order information, and locks it so that multiple field workers are not assigned to perform the same work. Then the application enters idle state until the application receives work order completion notification from the field worker; it follows to routine 2047, in which the application updates work order information according to the response from the field worker, and follows to routine 2048. In routine 2048 the system may send status reports to the field worker. Routine 2048 is followed by the "END" step, which concludes routine 2020.)

Regarding claim 11

Referring to claim 11 Mikhailov and Fascenda discloses all the limitations of claim 11 which is described above. Mikhailov also discloses wherein said each message is an extensible markup language package. (Col.11 lines 44 - 47 The document formats that may be supported by PAT are not restricted to a particular format and may include any data presentation format such as HTML, xHTML, XML, WML, SVG, etc.)

Regarding claim 12

Referring to claim 12 Mikhailov and Fascenda discloses all the limitations of claim 12 which is described above. Mikhailov also discloses receiving a polling request from said application server, said polling request establishing a transaction; and dequeing said each message from said queue and sending said each message toward said destination for said application of said application server in the context of said transaction. (Col.23 lines 61-67 and Col. 25 lines 1-3 If field worker accepts the work order, the "YES" branch is followed to routine 2046, in which the application updates the work order information, and locks it so that multiple field workers are not assigned to perform the same work. Then the application enters idle state until the application receives work order completion notification from the field worker; it follows to routine 2047, in which the application updates work order information according to the response from the field worker, and follows to routine 2048. In routine 2048 the system may send status reports to the field worker. Routine 2048 is followed by the "END" step, which concludes routine 2020.)

Regarding claim 13

Referring to claim 13 Mikhailov and Fascenda discloses all the limitations of claim 13 which is described above. Mikhailov also discloses receiving from said application server a message fro said mobile communication device; and forwarding said application server message to said wireless communication device. (Col. 61 FIG. 34B is a block diagram illustrating the invented application-specific device presence monitoring. The registration/deregistration request originates on the device as a result of routine 17000. It is submitted from the PAT 22 through Mobile Network(s) 25009, 25010 to Gateway(s) 25001, 25002, which will deliver the request to the Presence Monitor 1019 in the PES 27. The Presence Monitor 1019 checks the records according to algorithm 23000, and may communicate with Routing Tables 1032 to obtain device status before the request arrived. If the device status is changed, it will issue device presence update message to the Push Manager 1012, which through the Push Interface 1014 will publish the message to some Messaging System 25008, 25003, to a specific Topic or Queue 25012 the Application Server 25005, 25004 is subscribed to, through Subscription(s) 25004. Then Application Server 25005, 25004 receives the device network presence update message and may execute the application-specific logic. It is understood that the described process is an optional extension, which may be ignored by the applications not sensitive to device network presence.)

Regarding claim 14

Referring to claim 14 Mikhailov and Fascenda discloses all the limitations of claim 14 which is described above. Mikhailov also discloses wherein said pushing said each message toward said destination for said application of said application server comprising sending said message to a universal resource locator (URL). (Col. 21 lines 32-67 and Col.22 lines 1-32 The server system 1099 may include the following components: Proactivity Enablement Server (PES) 27, is a server software system, which facilitates server-initiated content delivery to mobile devices and is specially tailored for development and deployment of proactive wireless applications; Queue Storage 1020, is a supplement to PES, which stores content delivery request data for PES. It may be implemented using databases, files, etc.; Web Server 24, is a standard HTTP server or any other suitable application system, which is serving content requests and/or delivers information in the fixed network (Internet, Intranet, etc); Messaging System 1007, is a supplementary component, which function is to enable routing and delivery of digital messages in heterogeneous computing environment using well-defined messaging protocols. Examples of such messaging products include but not limited to Java Messaging Server (JMS) implementations, Microsoft MSMQ, Web Services SOAP protocol, and so forth; Application Server 25, is a system, which handles application logic and related code for proactive data applications 49 and submits content delivery requests to the Messaging System 1007 (L1); Database 26, is a component, which function is to store and manage application and other related data; other unlisted components, may be required to enable or extend functionality and/or performance of the system; PES 27 consists of the following modules: Push Interface 1014, is a communication interface to Messaging System 1007, which receives content delivery requests from Application Server 25 and delivers them to Content Manager 1021. It also obtains device presence notifications and content delivery status updates (L2) from Push Manager 1012 and publishes them to Messaging System 1007, which delivers the updates to the Application Server 25 (L3). Content Manager 1021, is a special communication component that interfaces with the Web Server 24, which upon receiving content delivery request containing content location information, such as URL, fetches the data from the Web Server 24, validates, parses, and supplies the content to Push Manager 1012 and Queue 46; Push Manager 1012, is the main engine for managing content delivery requests and sending out events on device presence to the Application Server 25. Through Router 1017, it supplies content to the device as defined in WAP Push, Push Access Protocol (PAP), or other applicable specifications.)

Regarding claim 17

Referring to claim 17 Mikhailov discloses all the limitations of claim 17 which is described above. Mikhailov also discloses a transaction server enabling use of at least one application server application by a wireless communication device, comprising :; a processor for, on receipt of a given message from said wireless communication device for a given application on said application server: queuing said given message on said a queue for said application; and subsequent to said queuing, pushing said given message, and each message queued on said queue, toward a destination for said application of said application server, and wherein said pushing by said processor comprises, for each message on said queue, dequeuing said message from said queue and pushing said each message. (Col. 61 Fig. 61 FIG. 34A is a block diagram illustrating the invented server-initiated content delivery process (see also FIG. 35A). The delivery request originates at the application server 25005, 25007, in response to application algorithms, that can be defined by those skilled in art at application design time. The request then follows to Messaging Systems 25003, 25008 to the messaging queue or topic that the application and PES 27 negotiated for content delivery requests. In this architecture the Application Server 25005, 25007 is the publisher of the messages to the messaging topics or queues 25012 and the PES 27 is the subscriber for the messages through message subscription(s) 25004. When the message arrives to the PES 27 it is received by Push Interface 1014, which provides bridging and decoding functions between Messaging System 25003, 25008 and PES 27 components. Once the message is processed the information follows to the Content Manager 1021, which fetches content delivery request data from the Application Servers 25005, 25007 or any other content provider using HTTP requests through Web Server 24 or other applicable means. Upon it receiving the data PES 27, preprocesses and validates it, and saves the content to the Queue 46, which stores the content in Queue storage 1020. In the next step the Content Manager 1021 communicates to the router 1017, which reads information from Routing tables 47, to decide if the device is available and can process server-initiated content delivery. Once the device is available, the content is sent to the PAT 22 through one or more gateways 25001, 25002 and Mobile Networks 25009, 25010. At this point the device receives the content and follows the algorithms described above to present the content to the user. During content delivery queuing and other related delivery processing Push Manager 1012 generates messages with delivery status notifications on each status change and publishes them through the Push Interface 1014 to messaging Queues or Topics 25012 in Messaging System 25003, 25008. Application Server 25005, 25007 can subscribe to the queues and topics through Subscriptions 25004 in order to obtain delivery status notifications if they are required by the application algorithms. Currently the PES 27 generates the messages for the following delivery events: The content is placed in the queue; The content is replaced in the queue (older content was suppressed to ensure delivery of the most fresh content only); The content delivery failed (with attempt number and error code), and delivery attempts will continue until queue for the frame expires; The content is delivered to the target device; The content expired (queue for the frame may be reset upon expiration of application-configured timeout) . Mikhailov did not disclose a memory storing at least one queue, with one queue being provided for each of said at least one application on said application server. The general concept of a memory storing at least one queue, with one queue being provided for each of said at least one application on said application server is well known in the art as taught by Fascenda. Fascenda discloses a memory storing at least one queue, with one queue being provided for each of said at least one application on said application server(Col. 10 lines 62-67 and Col. 11 lines 1-21 Sever template database 330 includes the latest or most current versions of all of the available templates in the system of the present invention. Server template database 330 also includes tables mapping individual client device unique identifiers to the most current template versions authorized for client devices 108 associated with the unique identifiers. As new services, features and options are added to the system of the present invention, new templates are stored in server template database 330 and/or existing stored templates are updated, to reflect the additions. Therefore, at any given time, it is possible a client device 108 includes an old version of a template, that is, an out-of-date template that requires updating. When server 114 receives information request message 316 from client device 108, server 114 determines whether the template (at client device 108) associated with the request message is the most current template (for example, an updated template). If the template is an old version, server 114 retrieves the most current template from server template database 330, and then transmits an appropriate template update, along with the requested information, to client device 108 using response message 318. In this manner, server 114 distributes the most current template versions to client devices 108 on a per access and an as needed basis. Thus, server 114 efficiently distributes template updates to client devices 108 to render new service features and options available to the users, and maintain configuration control over the clients.) It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Mikhialov to include a memory storing at least one queue, with one queue being provided for each of said at least one application on said application server in order to provide the user with a mechanism for retrieving the retained response information and to display such information as necessary.

Regarding claim 18

Referring to claim 18 Mikhailov and Fascenda discloses all the limitations of claim 18 which is described above. Mikhailov also discloses wherein said processor is further for, prior to said dequeuing and pushing, acquiring a lock for said destination on said application server, said lock preventing other of said destination. (Col.23 lines 61-67 and Col. 25 lines 1-3 If field worker accepts the work order, the "YES" branch is followed to routine 2046, in which the application updates the work order information, and locks it so that multiple field workers are not assigned to perform the same work. Then the application enters idle state until the application receives work order completion notification from the field worker; it follows to routine 2047, in which the application updates work order information according to the response from the field worker, and follows to routine 2048. In routine 2048 the system may send status reports to the field worker. Routine 2048 is followed by the "END" step, which concludes routine 2020.)

Regarding claim 19

Referring to claim 19 Mikhailov and Fascenda discloses all the limitations of claim 19 which is described above. Mikhailov also discloses wherein messages on each of said at least one queue are queued on a first in first out basis and wherein said processor is for refraining from pushing a trailing message in said queue until said processor considers a message in said queue immediately preceding said trailing message has successfully reached said destination. (Col. 61 Fig. 61 FIG. 34A is a block diagram illustrating the invented server-initiated content delivery process (see also FIG. 35A). The delivery request originates at the application server 25005, 25007, in response to application algorithms, that can be defined by those skilled in art at application design time. The request then follows to Messaging Systems 25003, 25008 to the messaging queue or topic that the application and PES 27 negotiated for content delivery requests. In this architecture the Application Server 25005, 25007 is the publisher of the messages to the messaging topics or queues 25012 and the PES 27 is the subscriber for the messages through message subscription(s) 25004. When the message arrives to the PES 27 it is received by Push Interface 1014, which provides bridging and decoding functions between Messaging System 25003, 25008 and PES 27 components. Once the message is processed the information follows to the Content Manager 1021, which fetches content delivery request data from the Application Servers 25005, 25007 or any other content provider using HTTP requests through Web Server 24 or other applicable means. Upon it receiving the data PES 27, preprocesses and validates it, and saves the content to the Queue 46, which stores the content in Queue storage 1020. In the next step the Content Manager 1021 communicates to the router 1017, which reads information from Routing tables 47, to decide if the device is available and can process server-initiated content delivery. Once the device is available, the content is sent to the PAT 22 through one or more gateways 25001, 25002 and Mobile Networks 25009, 25010. At this point the device receives the content and follows the algorithms described above to present the content to the user. During content delivery queuing and other related delivery processing Push Manager 1012 generates messages with delivery status notifications on each status change and publishes them through the Push Interface 1014 to messaging Queues or Topics 25012 in Messaging System 25003, 25008. Application Server 25005, 25007 can subscribe to the queues and topics through Subscriptions 25004 in order to obtain delivery status notifications if they are required by the application algorithms. Currently the PES 27 generates the messages for the following delivery events: The content is placed in the queue; The content is replaced in the queue (older content was suppressed to ensure delivery of the most fresh content only); The content delivery failed (with attempt number and error code), and delivery attempts will continue until queue for the frame expires; The content is delivered to the target device; The content expired (queue for the frame may be reset upon expiration of application-configured timeout)

Regarding claim 20

Referring to claim 20 Mikhailov and Fascenda discloses all the limitations of claim 20 which is described above. Mikhailov also discloses wherein said processor is further for, if a given message pushed from said given queue toward said destination does not successfully reach said destination, ceasing said dequeuing and pushing and re-queuing said given message on said given queue. (Col.62 Fig.35B lines 39-55 FIG. 35B is a logic flow diagram illustrating content loading routine 26004 of the server-initiated content delivery process 26000. This routine is typically implemented by the content manager 1021 in the PES 27. Routine 26004 starts with routine 26030, in which the content manager 1021 receives request to download certain content for the server-initiated content delivery request. Routine 26030 follows to routine 26031 in which the content manager 1021 loads the content from the URL read from the content delivery request and follows to the step 25033, in which it checks whether loading was successful. If loading was not successful, the "NO" branch is followed to routine 26043, in which the PES 27 sends the delivery failed message to the application server, which initiated content delivery request, through Messaging System 25008, 25003. If loading was successful, the "YES" branch is followed to the step 25035, in which the server checks if the content is of supported type. Here supported type is the type, which can be transformed by the gateway 23 to the type understood by the PAT 22.)

Regarding claim 21

Referring to claim 21 Mikhailov discloses a computer readable medium containing computer executable instructions for enabling use of an application server application by a wireless communication device, said computer executable instructions, when controlling a processor of a transaction server, causing said transaction server to: (Col. 61 FIG. 34B is a block diagram illustrating the invented application-specific device presence monitoring. The registration/deregistration request originates on the device as a result of routine 17000. It is submitted from the PAT 22 through Mobile Network(s) 25009, 25010 to Gateway(s) 25001, 25002, which will deliver the request to the Presence Monitor 1019 in the PES 27. The Presence Monitor 1019 checks the records according to algorithm 23000, and may communicate with Routing Tables 1032 to obtain device status before the request arrived. If the device status is changed, it will issue device presence update message to the Push Manager 1012, which through the Push Interface 1014 will publish the message to some Messaging System 25008, 25003, to a specific Topic or Queue 25012 the Application Server 25005, 25004 is subscribed to, through Subscription(s) 25004. Then Application Server 25005, 25004 receives the device network presence update message and may execute the application-specific logic. It is understood that the described process is an optional extension, which may be ignored by the applications not sensitive to device network presence.) Mikhailov did not disclose on receipt of a given message from said wireless communication device for said application on said application server, queue said given message on a queue for said application; and subsequent to said queuing, push said given message, each message queued on a queue for said application, toward a destination for said application of said application server, wherein said pushing comprises, for each message on said queue, deqeuing said each message from said queue and pushing said each message. The general concept of on receipt of a given message from said wireless communication device for said application on said application server, queue said given message on a queue for said application; and subsequent to said queuing, push said given message, each message queued on a queue for said application, toward a destination for said application of said application server, wherein said pushing comprises, for each message on said queue, deqeuing said each message from said queue and pushing said each message is well known in the art as taught by Fascenda. Fascenda discloses on receipt of a given message from said wireless communication device for said application on said application server, queue said given message on a queue for said application; and subsequent to said queuing, push said given message, each message queued on a queue for said application, toward a destination for said application of said application server, wherein said pushing comprises, for each message on said queue, deqeuing said each message from said queue and pushing said each message. (Col. 10 lines 62-67 and Col. 11 lines 1-21 Sever template database 330 includes the latest or most current versions of all of the available templates in the system of the present invention. Server template database 330 also includes tables mapping individual client device unique identifiers to the most current template versions authorized for client devices 108 associated with the unique identifiers. As new services, features and options are added to the system of the present invention, new templates are stored in server template database 330 and/or existing stored templates are updated, to reflect the additions. Therefore, at any given time, it is possible a client device 108 includes an old version of a template, that is, an out-of-date template that requires updating. When server 114 receives information request message 316 from client device 108, server 114 determines whether the template (at client device 108) associated with the request message is the most current template (for example, an updated template). If the template is an old version, server 114 retrieves the most current template from server template database 330, and then transmits an appropriate template update, along with the requested information, to client device 108 using response message 318. In this manner, server 114 distributes the most current template versions to client devices 108 on a per access and an as needed basis. Thus, server 114 efficiently distributes template updates to client devices 108 to render new service features and options available to the users, and maintain configuration control over the clients.) It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Mikhialov to include on receipt of a given message from said wireless communication device for said application on said application server, queue said given message on a queue for said application; and subsequent to said queuing, push said given message, each message queued on a queue for said application, toward a destination for said application of said application server, wherein said pushing comprises, for each message on said queue, deqeuing said each message from said queue and pushing said each message in order to provide the user with a mechanism for retrieving the retained response information and to display such information as necessary.

Conclusion

Arguments are deemed moot in view of the new grounds of rejection necessitated by the amendment.

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ashley D. Turner whose telephone number is 571-270-1603. The examiner can normally be reached on Monday thru Friday 7:30a.m.- 5:00p.m..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nathan J. Flynn can be reached on 571-272-1915. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Ashley D Turner
Examiner
Art Unit 2454

/Nathan J. Flynn/
Supervisory Patent Examiner, Art Unit 2454